Some Simple Self-Synchronizing Digital Data Scramblers

By J. E. SAVAGE

(Manuscript received July 28, 1966)

Two types of self-synchronizing digital data scramblers and descramblers are introduced and examined. The descramblers recover synchronization quickly after the insertion or deletion of channel bits, and they are relatively insensitive to channel errors. The scramblers act to increase the period of periodic data sequences, and the periodic channel sequences produced have approximately half as many transitions in one period as there are bits in a period. These circuits find application in common carrier systems where short-period data sequences produce high-level tones in the transmission band and, as a consequence, interchannel interference. And they have application when receiver clocks derive synchronization from transitions in the channel signal. A number of variations and modifications of the scramblers which affect their cost and size are considered.

The scramblers and descramblers are similar in construction and consist of linear sequential filters with either feed-forward or feedback paths, counters, storage elements and peripheral logic. The counters, storage elements and peripheral logic monitor the channel sequence but react infrequently so that the scramblers and descramblers behave principally as linear sequential filters.

I. INTRODUCTION AND SUMMARY

In this paper, we present two basic types of self-synchronizing digital data scramblers and descramblers. A scrambler is a digital machine which maps a data sequence into a channel sequence and, when the data sequence is periodic, into a periodic channel sequence with period which is many times the data period. When the source is periodic, the channel sequence produced by the scrambler also has many transitions.

A simple scrambler and one which is often used is a machine which adds a maximal-length shift-register sequence[1,2] to the data signal.

The scrambled data signal is then descrambled by the subtraction of the same maximal-length sequence. While this procedure is simple and easily implemented, it suffers from the serious disadvantage that the insertion or deletion of bits in the channel signal results in a descrambled sequence which is a garbled version of the data signal. The scramblers presented in this paper have the self-synchronizing property* and recover quickly from the insertion or deletion of channel bits.

There are two important applications for our scramblers. In common carrier systems small nonlinearities are present in modulators and demodulators which are used to frequency multiplex a bank of channels. Consequently, high-level tones in one channel may produce interference in other channels as a result of the nonlinearities in the mixing process. For this reason, systems engineers place limits on the levels of isolated tones in a customer's transmission band. Tones, in turn, are produced in digital data transmission systems by periodic data sequences and the limit on tone levels is then translated into a lower bound on the period of periodic channel sequences. Thus, our first application is to insure that any periodic source sequence is mapped by a scrambler onto a periodic channel sequence with sufficiently large period.

The second application for our scramblers concerns the need for transitions in the amplitude of the channel signal so that receiver clocks can derive bit or frame synchronization from the channel sequence. Receiver clocks often derive synchronization by passing the received signal through a filter tuned to the spectral component corresponding to the basic baud length and then observing the zero crossings of the filter output. Since the amplitude of the filter output will decrease to the background noise level if no transitions occur in the amplitude of the received signal or if the density of transitions is small, it is clear that in this application it is desirable to guarantee many closely spaced transitions in the amplitude of channel sequence when the source is periodic.

We introduce two basic types of self-synchronizing, digital data scramblers called multi-counter scramblers and single-counter scramblers and they are discussed in Sections IV and VI, respectively. Each scrambler consists of a "basic scrambler" and a "monitoring logic" which consists of additional storage elements, counters and incidental logic. We show in Section II that the "basic scrambler," which is a linear sequential filter with feedback paths and tap polynomial $h(x)$, responds to a periodic data sequence of period s by producing a periodic

* R. D. Fracassi and T. Tammaru introduced the self-synchronizing descrambler in a special scrambler for which they have a patent pending.[3]

channel sequence whose period is either s or the least common multiple of s and $p^m - 1$, where m is the number of stages in the basic scrambler and p is a prime greater than or equal to the number of elements in the source alphabet. The basic scrambler responds in this way to periodic inputs when its tap polynomial $h(x)$ is primitive over the modular field of p elements, $GF(p)$. The counters, logic, and storage elements of the monitoring logic monitor the channel sequence and respond whenever this sequence has as periods, one of the known data periods. The monitoring logic then reacts and changes the state of the basic scrambler, forcing it to have the long-period output.

We show in Sections IV and V that a multi-counter scrambler exists for binary as well as non-binary sources and we find the smallest thresholds required on counters in the monitoring logic of this scrambler. The single-counter scrambler is considered in Sections VI and VII and because of analytical difficulties we are only able to show the existence of this scrambler when the source is binary ($p = 2$) and the source periods are all prime to $2^m - 1$, where m is the size of the basic scrambler. Mixtures of the scramblers for binary sources are examined in Section VIII.

In Section IX we show that the scrambler output, when the input is periodic, contains many closely spaced transitions and that there are half as many transitions in one period as there are digits in that period. In Section XI we perform representative calculations to determine the spectrum of the scrambler output and find when the source is periodic that the output spectrum has P times as many tones as the unscrambled spectrum and each tone has $1/P$th the energy, where P is the factor by which the source period is increased.

The descramblers for each of the scramblers are discussed in the sections in which the scramblers are introduced and they are also discussed separately in Section X. In that section, we show that the descramblers recover synchronization rapidly after the insertion or deletion of channel digits and we observe that the principal effect of infrequent channel errors on the descramblers is to multiply the number of channel errors by $w(h)$, where $w(h)$ is the number of nonzero terms in the tap polynomial $h(x)$. In Section X we also note that the monitoring logics at the scrambler and descrambler reach threshold infrequently when the source is random and at most once when the source is periodic so that the descrambler monitoring logic may be removed and the descrambler considerably simplified as long as thresholding in the monitoring logic occurs at a tolerably low rate.

An example is given in Section XII of the application of the scramblers

and descramblers and representative calculations are performed to determine which scrambler configuration is least expensive. Section XIII closes with conclusions.

II. BASIC SCRAMBLER AND DESCRAMBLER

The shift register circuit shown in Fig. 1(a) is a linear sequential filter with feedhack paths[4] and is au example of the scrambling circuit which is hasic to the multi-counter scramhler and to the single-counter scramhler discussed in later sections. The linear sequential filter with feed-forward paths[4] shown in Fig. 1(h) is the complementary circuit to that shown in Fig. 1(a) and regenerates the data sequence from the channel sequence. We assume in these two examples that data is presented as a hinary sequence, that addition is taken modulo 2 and that the storage elements provide one hit of delay.

Examination of the circuits of Fig. 1 show that they have the required synchronization property since the effect of a hit lost or added in the line sequence is felt only as long as the values stored in the descramhler disagree with those stored at the scramhler, which is five hit intervals in our example.

A more general form for the hasic scramhler when the data is assumed to be a sequence of digits from the modular field of p elements, $GF(p) = \{0, 1, \cdots, p-1\}$, where p is prime, is shown in Fig. 2. Here, addition

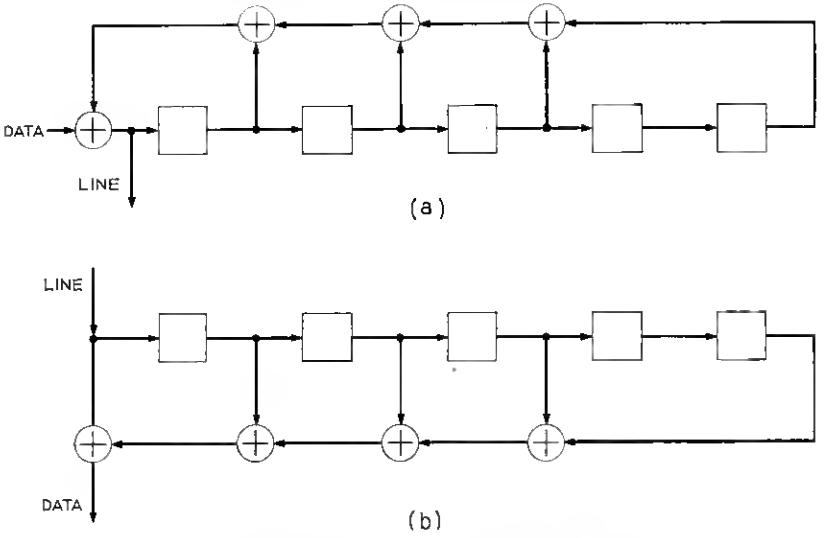

Fig. 1—(a) A basic scrambler; (b) a basic descrambler.

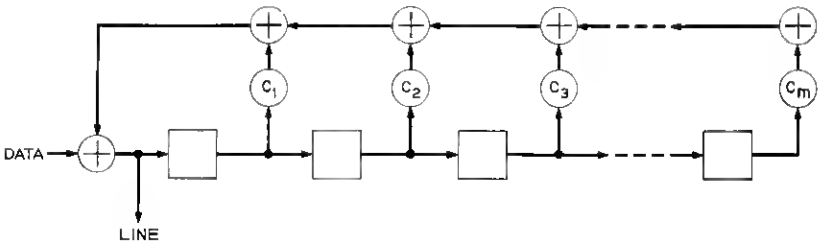

Fig. 2—General basic scrambler.

is taken modulo p and the outputs of the storage elements are multiplied by the tap constants $\{c_1, c_2, \cdots, c_m\}$ drawn from $GF(p)$. Here, multiplication is also taken modulo p. The tap constants must he constrained in a particular way if our scramblers are to extend the period of periodic sequences in the desired manner. Namely, the tap polynomial $h(x)$ in the indeterminate x given below

$$h(x) = x^m - c_1 x^{m-1} - \cdots - c_m \tag{1}$$

must be a primitive polynomial* over the field $GF(p)$. This condition will guarantee that the sequence generated by the basic scrambler in the absence of input will be either all zero or a maximal length sequence, that is a sequence which repeats but once every $p^m - 1$ digits. In the example given in Fig. 1, the tap polynomial is primitive over the binary field and it will generate a maximal length sequence of period $2^5 - 1 = 31$. ($h(x)$ is a primitive polynomial of degree m over the field $GF(p)$ if it is irreducible, that is, has no factors except 1 and itself, and if it divides $x^n - 1$ for $n = p^m - 1$ but does not divide it for any smaller n.)

Theorem 1: The basic scrambler described above when excited by a periodic sequence of period† s will respond with a periodic line sequence which has either period s or a period which is the least common multiple (LCM) of s and $p^m - 1$ ($LCM(s, p^m - 1)$). The period with which the scrambler responds is a function of the initial values stored in the scrambler storage elements, that is, its initial state, and there is but one such state (for each phase of the input sequence) for which the line sequence has period s. For all other such initial states the line sequence has the larger period.

This theorem is basic to all later results. It states that for only one starting state will the basic scrambler respond with period s to a data

* A nonprimitive polynomial may produce more than two output periods for an input period (see Theorem 1).

† A sequence will be said to be of period s if it has no smaller period.

sequence of that period. Thus, our objective, which is to extend the period of periodic sequences, is equivalent to detecting whether data preceding a periodic sequence has left the scrambler in the critical state for that sequence. Two basic methods of detecting the presence of the critical starting state when sequences of different periods are expected in the data are given in later sections.

III. PROOF OF THE BASIC SCRAMBLER THEOREM

Model a periodic input to the basic scrambler with a circulating register, as shown in Fig. 3 for an input of period 3. The initial state of the circulating register will be the first period of the periodic sequence. We let the vector $\mathbf{y}$ represent the state of the new circuit. Thus, if the input has period s and the basic scrambler has m stages, then $\mathbf{y}$ has $s + m$ components where the first s represent (in reverse order) the first period of the periodic input and the last m components represent the values stored in the corresponding storage elements when the periodic sequence begins. For example, $\mathbf{y} = (101101001)$ if the basic scrambler has the stored values 01001 when the sequence 1101, 1101, 1101, $\cdots$ arrives.

The circuit of Fig. 3 is linear since the next set of stored values is a linear combination of the preceding set. Thus, the state $\mathbf{y}'$ following $\mathbf{y}$ can be found by a matrix operation on $\mathbf{y}$ by the matrix T given below,* that is, $\mathbf{y}' = T\mathbf{y}$ where $\mathbf{y}$ and $\mathbf{y}'$ are taken to be column vectors.

$$T = \left[\begin{array}{ccc|ccccc} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right]. \tag{2}$$

For the general basic scrambler and an input of arbitrary period,

* For an excellent discussion of the matrix approach to linear sequential switching circuits see B. Elspas, The Theory of Autonomous Linear Sequential Networks, IRE Trans. Circuit Theory, *6*, pp. 45–60, 1959, which is reprinted in Ref. 13.

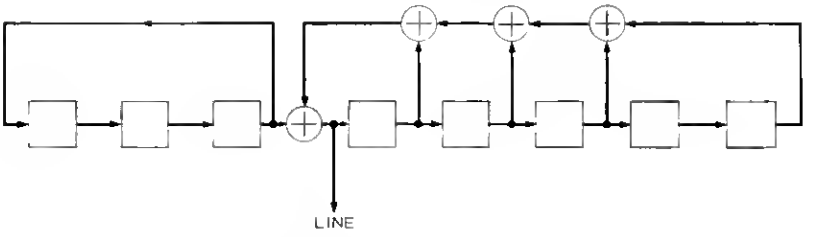

Fig. 3—Model of basic scrambler with periodic input.

say s, the matrix T has the following form

$$T = \left[\begin{array}{c|c} R & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \\ \hline 1 \; 0 \; \cdots \; 0 & T_h \end{array}\right], \tag{3}$$

where R is $s \times s$ and is shown below

$$R = \begin{bmatrix} 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & & \cdots & 0 \end{bmatrix}. \tag{4}$$

T_h is $m \times m$ and is given as

$$T_h = \begin{bmatrix} c_1 & c_2 & \cdots & c_m \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{5}$$

Since the state $\mathbf{y}'$ is found from $\mathbf{y}' = T\mathbf{y}$, all succeeding states are found by taking powers of T, that is, the ith state succeeding $\mathbf{y}$ is

$$\mathbf{y}_i = T^i\mathbf{y}. \tag{6}$$

The line sequence generated by a periodic input to the basic scrambler is periodic of the same period as the state $\mathbf{y}_i$ of the circuit which models the basic scrambler and periodic input. Thus, we prove Theorem 1 by studying the cycles of (6).

There is an indirect approach[5] that one can take to study the cycles

of (6). It amounts to a proof that these cycles are isomorphic to cycles of a matrix T^* obtained from T by eliminating the solitary 1 shown in (3). This amounts to disconnecting the circulating register from the basic scrambler and observing that the basic scrambler, which is a maximal length sequence generator,[6] has period 1 or $p^m - 1$, $m = \deg h(x)$, so that cycles of T^* have period s or $LCM(s, p^m - 1)$. The proof that the register can be disconnected amounts to showing that the minimal and characteristic polynomials of T are the same and equal those of T^*. Then, the elementary divisors of T and T^* are the same and their cycles are isomorphic.

Since there is a direct proof of Theorem 1 which contains many results important to the remainder of the paper, we present it here. If the basic scrambler with periodic input starts with state $\mathbf{y}$, then it has a cycle of length g if $T^g\mathbf{y} = \mathbf{y}$. The basic scrambler output will then be periodic with period g. We now ask for those values of g for which $T^g\mathbf{y} = \mathbf{y}$ has a solution. We begin by writing

$$\mathbf{y} = \mathbf{y}_s + \mathbf{y}_m , \tag{7}$$

where $\mathbf{y}_s$ is such that its first s components equal those of $\mathbf{y}$ and its last m components are zero. The vector $\mathbf{y}_m$ is zero in its first s components and is equal to $\mathbf{y}$ in its last m components. We can interpret $\mathbf{y}_m$ as the "starting state" of the basic scrambler and $\mathbf{y}_s$ as the state of the model for the periodically driven basic scrambler when the starting state is zero.

If

$$T^g\mathbf{y} = \mathbf{y} \tag{8}$$

then

$$-T^g\mathbf{y}_s + \mathbf{y}_s = T^g\mathbf{y}_m - \mathbf{y}_m \tag{9}$$

since T is a linear operator. We assume that the periodic input is fixed and has period which is strictly s. Then, the left-hand side of (9) is fixed and we ask whether a solution $\mathbf{y}_m$ for it exists for a given value of g. We note that

$$T^g = \begin{bmatrix} R^g & 0 \\ * & T_h^g \end{bmatrix} , \tag{10}$$

where the asterisk indicates some submatrix. Therefore, $T^g\mathbf{y}_m - \mathbf{y}_m$ is a vector whose first s components are zero. The left-hand side of (9) has its first s components zero only when g is a multiple of s because in that case $R^g = I_s$, the $s \times s$ identity matrix, and otherwise $R^g -$

$I_s \neq 0$ which means that a cyclic shift of the first s components of $\mathbf{y}_s$ when added to $\mathbf{y}_s$ is nonzero unless $g = ks$ for some integer $k \geqq 1$.

If we use the notation $(\mathbf{y})'$ to indicate the last m components of $\mathbf{y}$ we have for (9) the following when $g = ks$

$$(-T^{ks}\mathbf{y}_s + \mathbf{y}_s)' = [T_h^{ks} - I](\mathbf{y}_m)' \tag{11}$$

where I is now $m \times m$. We use the following theorem on (11).

Theorem 2: The matrix T_h has characteristic polynomial $h(x)$ which is assumed primitive over $GF(p)$. Therefore, $T_h^i - I$ is nonsingular for $i = 1, 2, \cdots, p^m - 2$ and $T^n = I$ for $n = p^m - 1$.

Proof: See appendix.

Since $T_h^n = I$ for $n = p^m - 1$, we can reduce ks modulo n so that T_h^{ks} can be written as a power of T_h less than n. In particular for $k < k_0$ where $k_0 s$ is the least commou multiple of s and n, which we call e, that is,

$$e = k_0 s = LCM(s, p^m - 1), \tag{12}$$

the matrices T_h^{ks} can be written as $T_h^{i_k}$ where $0 < i_k < n$. We have

$$T_h^{k_0 s} = T_h^e = (T_h^n)^{s/n} = (I)^{e/n} = I. \tag{13}$$

Returning to (11) we see that $T^{ks} - I$ is nonsingular for $1 \leqq k < k_0$. Therefore, when $k = 1$, (11) possesses a unique solution $\mathbf{y}_m$. That is, there exists a unique starting state $\mathbf{y}_m$ for each periodic sequence (modeled by $\mathbf{y}_s$) having period strictly s such that $T^s(\mathbf{y}_m + \mathbf{y}_s) = \mathbf{y}_m + \mathbf{y}_s$. Similarly, there exists a unique solution to (11) for each $2 \leqq k < k_0$. However, if $T^s\mathbf{y} = \mathbf{y}$, $\mathbf{y} = \mathbf{y}_m + \mathbf{y}_s$, then $T^{ks}\mathbf{y} = \mathbf{y}$ so that the cycles having period ks are really repetitions of the single cycle having period s. Also, when $k = k_0$, $T^{k_0 s} = I$ and $T^{k_0 s}\mathbf{y} = \mathbf{y}$ for all $\mathbf{y}$. We conclude that *for a prescribed input having period strictly s, the basic scrambler will respond with period s for only one starting state $\mathbf{y}_m$ and for all ather starting states will respond with period e given by* (12). This proves Theorem 1.

We have finished our discussion of the basic scrambler. We now consider the techniques used to detect the presence of a periodic sequence of low period on the line and present the first of two methods for altering the starting state of the basic scrambler. This first method is more general than the second and allows for the simultaneous detection of sequences of several periods. The second method applies only when the sequences expected on the line have periods which divide one of two numbers.

IV. THE MULTI-COUNTER SCRAMBLER

The general form of the multi-counter scrambler (MCS) is shown in Fig. 4. (The descrambler is shown in Fig. 5.) There are N counters, one for each period s_i, $1 \leqq i \leqq N$, and the ith counter will generate $+1$ if it reaches its threshold t_{s_i}. A counter is reset whenever the reset lead is nonzero so that t_{s_i} consecutive zeros on the reset lead of the ith counter will cause it to reach its threshold. All counter outputs are fed to the OR circuit shown so that a 1 is generated at the exclusive OR and added to the "tap sum"* whenever a counter reaches threshold. At the same time, all counters are automatically reset.

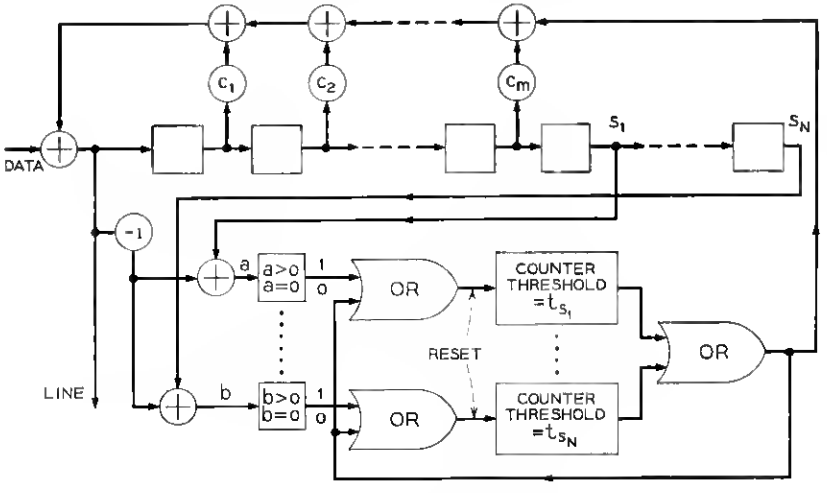

Fig. 4—Multi-counter scrambler.

The input to the ith counter is the difference between the present line digit and the digit transmitted s_i clock intervals earlier. If the line sequence has period s_i, then these two digits agree and the difference is zero. Then, the ith counter will reach threshold, the tap sum will be altered and the state of the basic scrambler changed.† The line sequence will then be changed from period s_i to period $LCM(s_i, p^m - 1)$ where p is the size of the modular field $GF(p)$ and m is the number of stages in the basic scrambler.

* We define the "tap sum" as the quantity added to the next data bit at the input to the basic scrambler.

† If the starting state of the basic scrambler is critical for a sequence of period s_i, then the state after j clock intervals is critical for the jth cyclic shift of the input sequence. Hence, a change in the tap sum will force the next state to be noncritical.

We observe, then, that the multi-counter scrambler for any choice of thresholds $\{t_{s_i}, 1 \leqq i \leqq N\}$ will force the basic scrambler to switch from a critical state to a noncritical state whenever the input has period $s_1, s_2, \cdots$, or s_N or some period which divides an s_i. It should be clear, however, that it is not necessary and perhaps not desirable to change the tap sum and the next state of the basic scrambler when the output does not have period s_i, $1 \leqq i \leqq N$, or some period which divides an s_i. The next theorem specifies the minimum values of the thresholds t_{s_i}, $1 \leqq i \leqq N$, so that the tap sum is changed only when "necessary." (Note that random data may generate line sequences which resemble periodic sequences and in such cases it will be "necessary" to change the tap sum.)

Theorem 3 (MCS Theorem): The multi-counter scrambler shown in Fig. 4 will scramble a periodic sequence of period s if s divides s_i for some i, $1 \leqq i \leqq N$, and will produce a periodic line sequence of period LCM $(s, p^m - 1)$ if the following two conditions are met:

(i) The tap polynomial $h(x)$ of degree m is primitive over $GF(p)$ where data sequences have components from $GF(p)$.

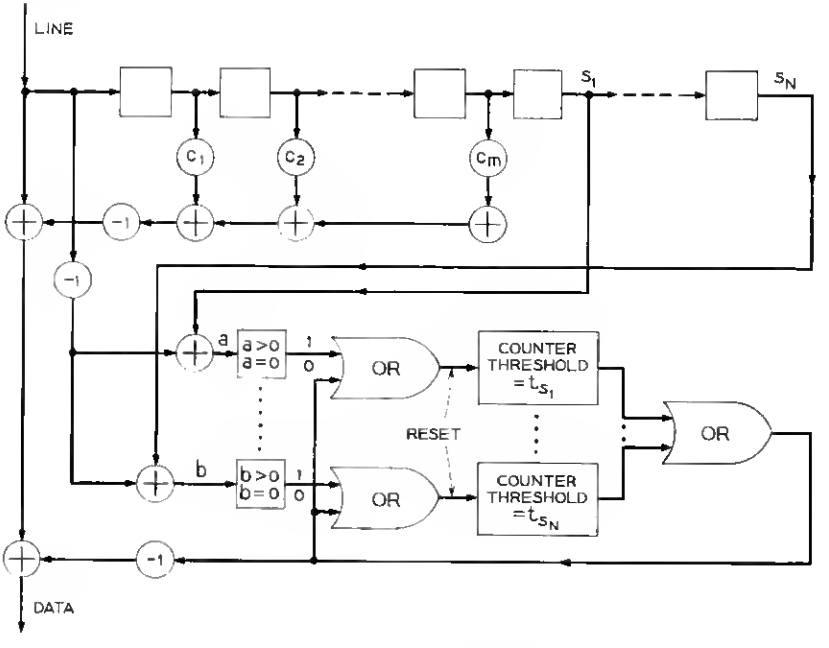

Fig. 5—Multi-counter descrambler.

(*ii*) *The thresholds* t_{s_i}, $1 \leqq i \leqq N$, *are chosen as*

$$t_{s_i} \geqq (m-1) + \max_{\substack{1 \leq j \leq N \\ j \neq i}} s_j .$$

If all input periods divide s_0, *then the theorem holds when condition* (*i*) *is met and a threshold of* $t_{s_0} \geqq m$ *is used.*

The descrambler for the MCS, shown in Fig. 5, has the self-synchronization property as long as line errors do not occur. When line errors occur, the counters in the descrambler may not read the same levels as the corresponding counters in the MCS. However, as seen from the MCS theorem, the counters must reset at least ouce every $(m-1) + \max_i s_i$ clock intervals when the input is periodic so that counter synchronization is easily established in this case. With random data the situation is not quite so clear. The question of descrambler synchronization, including the effect of channel errors, is considered in detail in Section X.

V. PROOF OF THE MCS THEOREM

We use the notation developed in Section III for the proof of Theorem 3. Fig. 6 shows the basic scrambler with periodic input of period s and one counter. The only input to the basic scrambler other than the data input is the lead from the counters which is used to change the tap sum. The counter shown is assigned to the detection of line sequences whose periods divide s_i.

To prove Theorem 3 we must show that t_{s_i} can be chosen such that

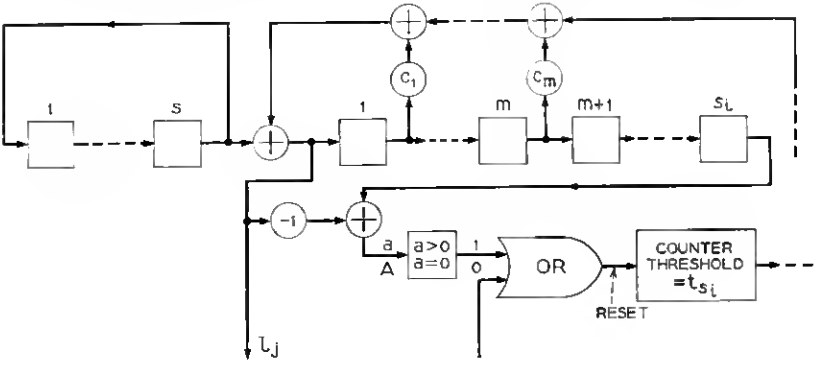

Fig. 6—One counter of the MCS with periodic input.

the ith counter does not reach thresbold when the data sequence is periodic unless the line sequence has period s_i or some period wbich divides s_i. Before we begin our proof we introduce some notation. In Fig. 6, we use l_j to indicate the jth line digit calculated with data from a periodic input of period s. The basic scrambler is sbown and we indicate with the vector $\mathbf{y}$ the state of the linear sequential filter composed of the circulating register of s stages and the m stages of the basic scrambler. Call this filter of $s + m$ stages the driven basic scrambler. Then, from (6) the next state of the driven basic scrambler, $\mathbf{y}'$, is

$$\mathbf{y}' = T\mathbf{y} \tag{8}$$

provided that the monitoring logic is not active. If it is active, that is, if one or more counters reach threshold, tben

$$\mathbf{y}' = T\mathbf{y} + \mathbf{y}_t \tag{14}$$

where $\mathbf{y}_t$ contains a single one in its $(s + 1)$th position.

The first line digit calculated with the periodic input, l_1 , is

$$l_1 = [T\mathbf{y} + u_1\mathbf{y}_t]_s \tag{15}$$

where

$$[\mathbf{z}]_s = z_{s+1} , \tag{16}$$

the $(s + 1)$th component of the vector z, and

$$u_1 = \begin{cases} 1 & \text{monitoring logic active at first calculation,} \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

In general, the jth line digit is

$$l_j = \left[T^j\mathbf{y} + \sum_{k=1}^{j} u_k T^{j-k}\mathbf{y}_t \right]_s \tag{18}$$

where

$$u_k = \begin{cases} 1 & \text{monitoring logic active at } k\text{th calculation,} \\ 0 & \text{otherwise.} \end{cases} \tag{19}$$

Now consider the sequence $\{a_j\}$ calculated at point A of Fig. 6. If a run of consecutive zeros in this sequence is large enough, the ith counter will reach thresbold unless some other counter reaches thresbold before it. When the periodic input begins, the counters in the MCS will be at unknown levels and the $(\max_i s_i)$ stored values will be, in

general, unrelated to the input; thus one or more counters may reach threshold before l_1 reaches the s_ith storage element of the MCS.* We now wish to show that *the sequence* $\{a_j\ ,\ j \geqq s_i + 1\}$ *will contain a run of no more than* $\max_{j \neq i} (m - 1 + s_j)$ *zeros if the line sequence is periodic with a period which does not divide* s_i .

We have for $j \geqq s_i + 1$

$$a_j = -l_j + l_{j-s_i} \tag{20}$$

and

$$a_j = -\left[T^{j-s_i}(T^{s_i}\mathbf{y} - \mathbf{y}) + \sum_{k=1}^{j} u_k T^{j-k}\mathbf{y}_t - \sum_{k=1}^{j-s_i} u_k T^{j-s_i-k}\mathbf{y}_t\right]_s . \tag{21}$$

If $u_{s_i+1} = 0$ then let j_0 be such that $u_j = 0$, $s_i + 1 \leqq j < j_0$ and $u_{j_0} = 1$, that is, the most recent thresholding occurs at $j = j_0$. (The case $u_j = 0$ for all $j \geqq s_i + 1$ is trivial, so we assume that $u_{j_0} = 1$ for some j_0 .) Then, if we write $\mathbf{z}_i$ as

$$\mathbf{z}_i = \sum_{k=1}^{j_0} u_k T^{j_0-k}\mathbf{y}_t - \sum_{k=1}^{j_0-s_i} u_k T^{j_0-s_i-k}\mathbf{y}_t \tag{22}$$

and if we ignore all counters except the ith, we have

$$a_j = -[T^{j-j_0}(T^{j_0-s_i}\{T^{s_i}\mathbf{y} - \mathbf{y}\} + z_i)]_s \tag{23}$$

for $j_0 \leqq j \leqq j_0 + t_{s_i} - 1$. For this range of j the a_j can be viewed as the values appearing in the $(s + 1)$th storage element of the driven basic scrambler with starting state $\mathbf{y}_i^*$

$$\mathbf{y}_i^* = T^{j_0-s_i}\{T^{s_i}\mathbf{y} - \mathbf{y}\} + z_i . \tag{24}$$

Now, assume that input period s does not divide s_i . Then, the first s components of $T^{j_0-s_i}\{T^{s_i}\mathbf{y} - \mathbf{y}\}$ are not all zero. Since z_i is zero in its first s components, the starting state $\mathbf{y}_i^*$ is nonzero in some of its first s components. Consequently, the state of the driven basic scrambler (of $s + m$ stages) can never be completely zero† so that the sequence $\{a_j\ ,\ j \geqq j_0\}$ cannot contain more than $s + m - 1$ consecutive zeros if s does not divide s_i .‡ We shall now show that, in fact, the sequence $\{a_j\ ,\ j \geqq j_0\}$ cannot contain more $s + m - 2$ consecutive zeros if $s \nmid s_i$. We shall also show that there exists an input of period s if $s_i = ks \pm 1$

---

* Note that the lead from the counters will be active at most once during the first s_i calculations if $s_i \leqq \min_j t_{s_j}$.

† Note that the matrix operator T just circulates the first s components of $\mathbf{y}^*$.

‡ We will use the notation $s \nmid s_i$ to mean s does not divide s_i .

for some integer $k \geqq 1$ such that the sequence $\{a_j, j \geqq j_0\}$ will have this many consecutive zeros.

In (24) the first s components of $\mathbf{y}_i^*$ are a cyclic shift of the first s components of $T^{s_i}\mathbf{y} - \mathbf{y}$. Recalling the definition of T from (3) we see that these s components are the components of the vector $(R^{s_i} - I)(\mathbf{y}_s)''$ where R is $s \times s$ and is given by (4) and $(\mathbf{y}_s)''$ is the vector containing the first s components of $\mathbf{y}_s$. The vector $(R^{s_i} - I)(\mathbf{y}_s)''$ cannot contain a single nonzero element if $s \nmid s_i$ as seen below by example: Let $s = 4$, $s_i = 5$ and $(\mathbf{y}_s)$ have components y_1, y_2, y_3, y_4. Then, we have

$$(R^{s_i} - I)(\mathbf{y}_s)'' = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} b \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{25}$$

where b is the single nonzero element. Thus,

$$\begin{aligned} -y_1 + y_3 &= b \neq 0 \\ -y_2 + y_4 &= 0 \\ +y_1 - y_3 &= 0 \\ +y_2 - y_4 &= 0. \end{aligned} \tag{26}$$

It is clear that two equations, the first and third, cannot both be satisfied. This will be true regardless of the location of the single nonzero element. Hence, *there must be at least two nonzero elements in the first s components of* $\mathbf{y}_i^*$. Consequently, $\{a_j, j \geqq j_0\}$ cannot contain a run of more than $s + m - 2$ consecutive zeros. If $\mathbf{y}_s$ contains a single nonzero element and if $s_i = ks \pm 1$ then $(R^{s_i} - I)(\mathbf{y}_s)''$ will contain two consecutive nonzero elements. Also, since $\mathbf{y}_m$ is in general arbitrary, it can be chosen so that the first $s + m - 2$ digits generated with $\mathbf{y}_i^*$ as the starting state will be zero.

At this point, we have shown that a periodic input of period s, where $s \mid s_j$, some $j \neq i$ but $s \nmid s_i$, will not cause the ith counter to reach threshold more than once after the s_ith line digit is transmitted if we choose t_{s_i} to be

$$t_{s_i} = (m - 1) + \max_{j \neq i} s_j . \tag{27}$$

This is true since the sequence generated at point A of Fig. 6 will not show more than t_{s_i} consecutive zeros after the first time the monitoring

logic is active following the transmission of the s_ith line digit. We also note that a threshold of the size given above may be necessary if there exists an s such that $s_i = ks \pm 1$, some $k \geqq 1$, where $s \mid s_j$, some j.

Consider next the case where $s \mid s_j$. If the line sequence is periodic of period s (see Theorem 1) at any time after the periodic sequence begins, the sequence at point A will contain an indefinite number of zeros so that the ith counter will definitely reach threshold (unless $s \mid s_j$, some $j \neq i$, and $t_{s_j} < t_{s_i}$, in which case the jth counter may reach threshold first). Since there is only one critical state for each periodic sequence, the change in the tap sum resulting from the detection of the period s line sequence will cause the output to have period $LCM(s, p^m - 1)$. In this case the vector $\mathbf{y}_i^*$ in (24) cannot be entirely zero (its first s components are zero, however) because it would then result in an all zero sequence at point A. Thus, the last m components of $\mathbf{y}_i^*$ must contain at least a single nonzero component. But $[T^{j-i_0}\mathbf{y}_i^*]_1$ (which generates $\{a_j, j \geqq j_0\}$) then is just the output of a maximal length sequence generator (see appendix) so that no more than $m-1$ consecutive zeros will be seen at point A if $s \mid s_i$ and the output has period $LCM(s, p^m - 1)$.

In conclusion, *if* $s \mid s_i$ *but the output does not have period* s *or if* $s \nmid s_i$ *but* $s \mid s_j$ *some* $j \neq i$, *then the* i*th counter will reach threshold at most once after the transmission of the* s_i*th line digit if the* i*th threshold* t_{s_i} *is chosen as*

$$t_{s_i} = (m - 1) + \max_{j \neq i} s_j \,. \tag{28}$$

Of course, the same is true for any threshold larger than t_{s_i}.

VI. THE SINGLE-COUNTER SCRAMBLER

The single-counter scrambler (SCS) is shown in Fig. 7 (and the descrambler is shown in Fig. 8). This scrambler is designed to scramble periodic *binary* sequences whose periods divide either s_1 or s_2 or both. It has a single counter and for some applications may be less costly to build than the multi-counter scrambler. And while we consider the SCS when the input periods divide either s_1 or s_2 or both, one may be able to design for the case of many more input periods.

The SCS has two circuits for detecting periodic sequences. If either or both of the two detecting circuits produces 0 at any one time, one cannot with a single measurement determine whether the line sequence has period s_1 or s_2. On the contrary, if both circuits produce a nonzero

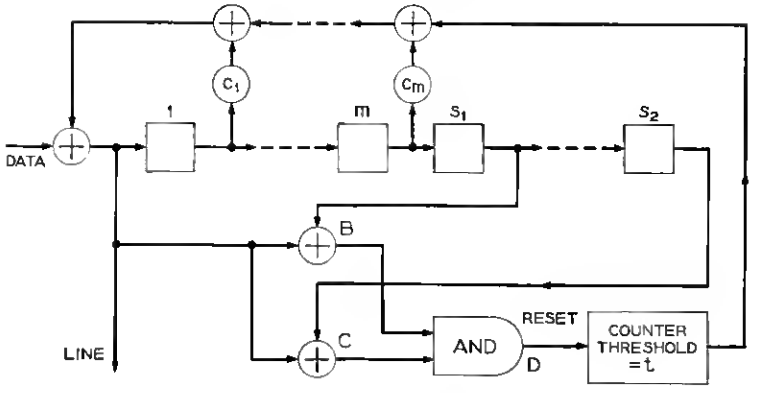

Fig. 7—Single counter scrambler.

output, it is clear that the line sequence does not have period s_1 or s_2 and that the counter should be reset. A 2-input AND gate has a nonzero output only when both inputs are nonzero, consequently, we use it as input to a counter, as shown in Fig. 7. This counter will reach threshold after t line transmissions if each of t consecutive pairs of outputs of the detecting circuits contains one or more 0's.

The major design problem of the SCS is the choice of the counter threshold. This is not an easy problem, unfortunately, and all that we

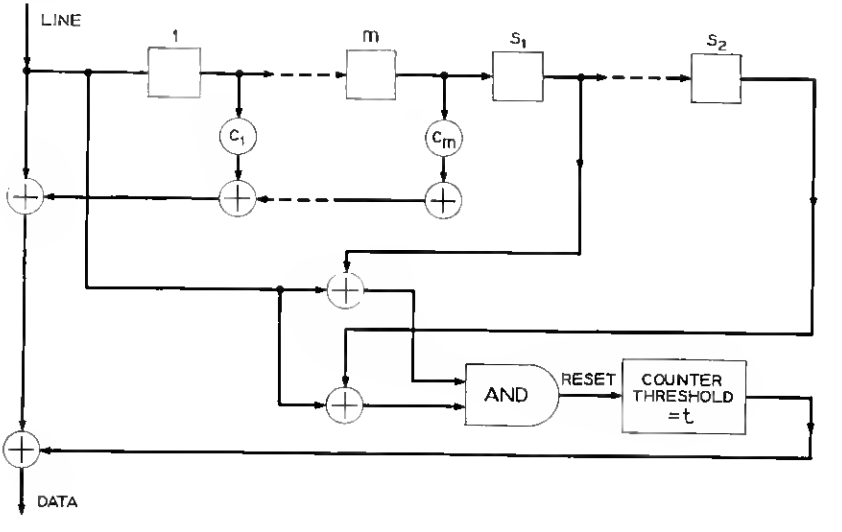

Fig. 8—Single counter descrambler.

have been able to say about it is that counter thresholds do exist when $p = 2$ (the source is binary) and the input periods are relatively prime to $2^m - 1$ and then to only give a gross upper bound on the smallest permissible threshold. The following theorem states what is known about the threshold for the SCS.

Theorem 4 (SCS Theorem): A single-counter scrambler which will scramble all periodic binary sequences with periods which divide s_1 or s_2 ($s_1 < s_2$, $s \nmid s_2$) exists if

(i) *the tap polynomial $h(x)$ of degree m is primitive over $GF(2)$,*
(ii) *s_1 and s_2 are relatively prime to $2^m - 1$, and*
(iii) *a counter threshold, t, $t \leqq s_2(2^m - 1) - 2^{m-1} + 2$ is chosen.*

This theorem does not rule out the possibility that an SCS exists when s_1 and s_2 are not both relatively prime to $2^m - 1$ nor does it rule out an SCS for nonbinary data. It simply states that when conditions (i) and (ii) are met, one can show that a counter with threshold t, $t \leqq s_2(2^m - 1) - 2^{m-1} + 2$, will not reach threshold when the output of the basic scrambler has period $s \times (2^m - 1)$ where s divides s_1 or s_2 or both. In fact, the bound on the threshold required to prevent the counter from reaching threshold prematurely is many times larger than necessary. In the example given in Section XII the bound is more than 35 times too large.

VII. PROOF OF THE SCS THEOREM

For the proof of Theorem 4, we recall the proof of Theorem 3. In particular, it is instructive to review the discussion surrounding equations (20) through (24). We recall that a_j of (20) is the jth digit calculated (after the arrival of the periodic data sequence) at the input to the ith counter of Fig. 6. We argued that if the ith counter reaches threshold on the j_0th calculation, $j_0 \geqq s_i + 1$, then a_j could be calculated from

$$a_j = -[T^{j-j_0}\mathbf{y}_{j_0}^*], \tag{29}$$

for $j \geqq j_0$ and until the next time the ith counter reaches threshold. Here $[\mathbf{y}_s]$ indicates the $(s+1)$st component of $\mathbf{y}$ and $\mathbf{y}_{j_0}^*$ is given by (24). Thus, the sequence generated at point A of Fig. 6, namely $a_{j_0}, a_{j_0+1}, \cdots$ can be viewed as generated by the basic scrambler with periodic input and $\mathbf{y}_{j_0}^*$ as starting state.

In Theorem 4 we assume that the data sequence is binary so that the above equations apply if we interpret subtraction as addition since

they are equivalent on the binary field. In Fig. 7 the sequences $\{b_j\}$ and $\{c_j\}$ are generated at points B and C, respectively. We wish to show that the largest run of consecutive zeros in the logical AND of $\{b_j\}$ and $\{c_j\}$ after a certain transient period cannot exceed $s_2(2^m - 1) - 2^{m-1} + 1$ when $s_1 < s_2$ and s_1 and s_2 are both relatively prime to $2^m - 1$.

Consider the sequences $\{b_j\ ,\ j \geqq s_2 + 1\}$ and $\{c_j\ ,\ j \geqq s_2 + 1\}$. Then, if the counter reaches threshold at* $j_1\ ,\ j_1 \geqq s_2 + 1$, these two sequences will cause the counter to reach threshold only once more if the line sequence has period s, $s \mid s_1$ or $s \mid s_2$ or both. If the line sequence has period† $s \times (2^m - 1)$, then neither $\{b_j\ ,\ j \geqq j_1\}$ nor $\{c_j\ ,\ j \geqq j_1\}$ can be all zero since this would imply that the line sequence has period s_1 or s_2 .

We now consider two cases, case I when s divides both s_1 and s_2 and case II when s divides s_1 but not s_2 or vice versa. From (24) it is clear that in case I both $\{b_j\ ,\ j \geqq j_1\}$ and $\{c_j\ ,\ j \geqq j_1\}$ are the outputs of basic scramblers with no input and with nonzero starting states so that they repeat with period $2^m - 1$. In case II when $s \mid s_1$, say, but not s_2, $\{b_j\ ,\ j \geqq j_1\}$ is the output of a basic scrambler with nonzero starting state and has period $2^m - 1$ while $\{c_j\ ,\ j \geqq j_1\}$ is the output of a driven basic scrambler with input period s. (The input may not be strictly of period s, however, as we shall see later.)

The logical AND of the sequences generated at points B and C of Fig. 7 can be interpreted as the sequence generated by the normal arithmetic multiplication of b_j and c_j . Thus, the sequence at point D of Fig. 7 has period $2^m - 1$ in case I and period $s \times (2^m - 1)$ in case II. Let $\mathbf{B}_n$ and $\mathbf{C}_n$ be n component vectors with $B_l = b_{j_1+l}$ and $C_l = c_{j_1+l}$. Then, at point D, the vectors $\mathbf{B}_n$ and $\mathbf{C}_n$ generate the n-vector $\mathbf{D}_n = \mathbf{B}_n \cdot \mathbf{C}_n$ where multiplication of $\mathbf{B}_n$ and $\mathbf{C}_n$ is term-by-term, i.e.,

$$\mathbf{D}_n = (B_1C_1\ ,\ B_2C_2\ ,\ \cdots\ ,\ B_nC_n). \tag{30}$$

Let $w(\mathbf{y}_n)$ be the Hamming weight[7] of the n-vector $\mathbf{y}_n$, that is, the number of 1's in $\mathbf{y}_n$. Then we have[8]

$$w(\mathbf{D}_n) = w(\mathbf{B}_n \cdot \mathbf{C}_n) = \frac{w(\mathbf{B}_n) + w(\mathbf{C}_n) - w(\mathbf{B}_n + \mathbf{C}_n)}{2}, \tag{31}$$

where addition is modulo 2. We now wish to use this last equation to find a lower bound on the number of 1's $\mathbf{D}_n$. From this we can obtain an upper bound on the number of consecutive 0's in D_n and an upper

* It may indeed reach threshold for $1 \leqq j_1 \leqq s_2$ but this does not affect our analysis.

† s_1 and s_2 are relatively prime to $2^m - 1$ so that the line sequence has period $s \times (2^m - 1)$ if $s \mid s_1$ or $s \mid s_2$ as seen from Theorem 1.

hound to the threshold required to prevent the counter of Fig. 7 from making unnecessary changes in the tap sum.

In case I we let $n = 2^m - 1$, which is the period of the sequence generated at point D. Thus, $\mathbf{D}_n$ is one period of this sequence. $\mathbf{B}_n$ and $\mathbf{C}_n$ are each one period of the output of the hasic scramhler (which is a maximal length sequence generator). Thus, $\mathbf{B}_n$ can be ohtained from $\mathbf{C}_n$ hy a cyclic shift and they both have the same Hamming weight. Then, we have the following result.

Lemma 1: If $n = 2^m - 1$ *and* $\mathbf{B}_n$ *and* $\mathbf{C}_n$ *are periods of a maximal length sequence with* $\mathbf{B}_n = \mathbf{C}_n$, *then* $w(\mathbf{D}_n) = w(\mathbf{C}_n) = 2^{m-1}$. *If* $\mathbf{B}_n \neq \mathbf{C}_n$, *then* $w(\mathbf{D}_n) = w(\mathbf{C}_n)/2 = 2^{m-2}$.

Proof: We need only show that $w(\mathbf{C}_n) = 2^{m-1}$. From the comments at the end of the appendix we have that the state of the autonomous basic scramhler, as a binary m-tuple, ranges through all $2^m - 1$ nonzero hinary m-tuples. Since the first digit of eaeh m-tuple is a line digit, there will he exactly 2^{m-1} 1's in one period of the line sequence generated by the autonomous hasic scrambler. (Note that the scramhler does not start with the zero state.) Q.E.D.

In case I, then, the number of consecutive zeros in the sequence at D cannot exceed $2^m - 1 - 2^{m-2}$ and a threshold of $2^m - 2^{m-2}$ will guarantee unnecessary tap sum changes in this case.

Consider now case II where $\{b_i, j \geqq j_1\}$ has period $2^m - 1$ and $\{c_i, j \geqq j_1\}$ is the output of a driven basic scrambler characterized by

$$c_i = [T^{i-j_1}\mathbf{y}_0^*]_s , \tag{32}$$

where $\mathbf{y}_0^*$ is an $(s + m)$-vector which from (24) has the form

$$\mathbf{y}_0^* = T^{j_1-s_2}\{T^{s_2}\mathbf{y} + \mathbf{y}\} + \mathbf{z} \tag{33}$$

where $\mathbf{y}$ is an arhitrary $(s + m)$-vector, except that its first s components model a periodic sequence of strictly period s, and $\mathbf{z}$ is zero in its first s components and arbitrary in its last m components. The first s components of $T^{s_2}\mathbf{y} + \mathbf{y}$ cannot be all zero if $s \nmid s_2$. It may model a periodic sequence of period s_0, however, where $s_0 < s$ and $s_0 \mid s$. In particular, we may have $s_0 = 1$ in which case the first s components of $\mathbf{y}_0^*$ may be 1's and $\{c_i, j \geqq j_1\}$ may have an output of period 1 consisting of the all 1 sequence. If this is true $\mathbf{D}_n = \mathbf{B}_n \cdot \mathbf{C}_n = \mathbf{B}_n$ and $\mathbf{D}_n$ will have no more than $m - 1$ consecutive zeros. If $\{c_i, j \geqq j_1\}$ has period $s_0 > 1$ it will contain no less than a single nonzero component in each period nor no more than $s_0 - 1$ nonzero components in each period.

Thus, if $n = s_0$, $s_0 > 1$, and $\mathbf{C}_n$ represents one period in the output of period s_0, we have

$$1 \leqq w(\mathbf{C}_n) \leqq (s_0 - 1). \tag{34}$$

When $\{c_j, j \geqq j_1\}$ has period $s_0 \times (2^m - 1)$ let $n = s_0 \times (2^m - 1)$ in (31). We now show that for this case

$$s_0(2^{m-1} - 1) \leqq w(C_n) \leqq s_0 \times 2^{m-1}. \tag{35}$$

The state of the driven basic scrambler with input of period s_0 can be represented with an $(s_0 + m)$-vector. There are $s_0 \times 2^m$ admissible state vectors since the last m components are arbitrary and the first s_0 components must be a cyclic shift of the first s_0 components of some other state vector. The last m components of these $s_0 \times 2^m$ vectors range through each of the 2^m m-tuples s_0 times. Since the $(s + 1)$st component of each state vector is a line digit $w(\mathbf{C}_n) \leqq s_0 \times 2^{m-1}$ which is the number of 1's shown in these positions. We also have $w(\mathbf{C}_n) \geqq s_0 \times 2^{m-1} - s_0$ since the components of $\mathbf{C}_n$ are generated by only $s_0 \times (2^m - 1)$ of the $s_0 \times 2^m$ admissible state vectors and the missing state vectors may all contain 1 in the $(s + 1)$st components.

Returning to (31) we see that the vector $\mathbf{B}_n + \mathbf{C}_n$ appears. It represents the first n components of $\{b_j + c_j, j \geqq j_1\}$. This is the output sequence of a driven basic scrambler driven with period s_0 and which has as a starting state the state which produces $\{c_j, j \geqq j_1\}$ and which is modified by the addition in its last m components of the starting state of the autonomous basic scrambler which produces $\{b_j, j \geqq j_1\}$. Since this last state is arbitrary, $\mathbf{B}_n + \mathbf{C}_n$ can be expected to have period s_0 or $s_0 \times (2^m - 1)$ and the bounds on the weight of $\mathbf{C}_n$ for these two periods apply to $\mathbf{B}_n + \mathbf{C}_n$.

We now combine our bounds with (31) to obtain a lower bound to $w(\mathbf{D}_n)$ for case II. Remember that $n = s_0(2^m - 1)$.

(*i*) Let $\mathbf{C}_n$ have period s_0, then $\mathbf{B}_n + \mathbf{C}_n$ has period $s_0 \times (2^m - 1)$ and

$$w(D_n) \geqq \frac{s_0 \times 2^{m-1} + (2^m - 1) - s_0 \times 2^{m-1}}{2} = \frac{2^m - 1}{2} \tag{36}$$

where $w(B_n) = s_0 \times 2^{m-1}$ from Lemma 1.

(*ii*) Let $\mathbf{C}_n$ have period $s_0 \times (2^m - 1)$ and $\mathbf{B}_n + \mathbf{C}_n$ have period s_0. Then

$$w(D_n) \geqq \frac{s_0 \times 2^{m-1} + s_0(2^{m-1} - 1) - (s_0 - 1)(2^m - 1)}{2} = \frac{2^m - 1}{2}. \tag{37}$$

(*iii*) Let $\mathbf{C}_n$ and $\mathbf{B}_n + \mathbf{C}_n$ have period $s_0(2^m - 1)$. Then

$$w(\mathbf{D}_n) \geqq \frac{s_0 \times 2^{m-1} + s_0(2^{m-1} - 1) - s_0 \times 2^{m-1}}{2} \geqq \frac{s_0}{2}(2^{m-1} - 1). \quad (38)$$

Therefore, the number of 1's in the sequence D_n of $n = s_0 \times (2^m - 1)$ components for case II must exceed $(2^m - 1)/2 - \frac{1}{2}$ and the number of consecutive zeros cannot exceed $s_0 \times (2^m - 1) - (2^m - 1)/2 + \frac{1}{2}$.

Combining the results for cases I and II we find that the number of consecutive zeros at point D of Fig. 7 when $s_1 < s_2$, $s_1 \nmid s_2$ and the input has period s, $s \mid s_1$ or $s \mid s_2$ or both, will not exceed $s_2(2^m - 1) - (2^m - 1)/2 + \frac{1}{2}$ unless the line sequence has period s. The threshold then need not be any larger than $s_2(2^m - 1) - 2^{m-1} + 2$ to prevent unnecessary changes in the tap sum. Q.E.D.

VIII. MIXTURES OF THE SCRAMBLERS

The two types of scramblers given above are distinguished by the structure of their monitoring logics. The MCS has one counter for each of the input periods $s_1, s_2, \cdots s_N$ and the SCS has a single counter to detect the presence of one of two periods, s_1 or s_2. We have found the smallest threshold required on each counter of the MCS so that they change the tap sum only when necessary. Also, we have shown the existence of a finite threshold on the single counter of the SCS when the source is binary and input periods are relatively prime to $2^m - 1$, where m is the number of stages in the basic scrambler.

Since the monitoring logic for both counters acts to detect the presence of periodic sequences of known periods in the line sequence, it sbould be clear that a monitoring logic eontaining a mixture of the MCS logic and the SCS logic may be used. We know of an SCS monitoring logic only when the source is binary, however, so that the mixture must be restricted to the binary source case. Thus, we may now consider a scrambler with a monitoring logic, a portion of which has counters detecting the presence of one of a pair of periods and another portion consisting of individual counters for single periods. The outputs of all counters are fed to an OR gate which in turn is added modulo 2 to the tap sum. The output of the OR gate is also used to reset all counters.

IX. TRANSITIONS IN A SCRAMBLED SEQUENCE

The basic scramblers described above may have applications in situations where bit framing at the receiver is derived from transitions

in the line signal. In this section we show that transitions occur frequently in a scrambled periodic sequence and that in one period of a scrambled sequence there are approximately half as many transitions as there are digits. These results are shown when the source is binary and the scrambler input periods are relatively prime to $2^m - 1$, where m is the size of the basic scrambler.

Let $\boldsymbol{l}$ represent one period of the line sequence generated by the basic scrambler when the input has period s. If the source is binary, if the basic scrambler has m stages and if s is relatively prime to $2^m - 1$, then $\boldsymbol{l}$ is an $s(2^m - 1)$ component vector. If we assume that the binary line sequence is converted into a line signal by the mapping $1 \rightarrow 1\ 0 \rightarrow -1$, and if it is linearly modulated, then transitions in the channel signal occur whenever transitions in the line sequence appear. Thus, we should like to know the number of transitions in $\boldsymbol{l}$ and the maximum separation between transitions.

Theorem 5: The binary vector $\boldsymbol{l}$ of length $s(2^m - 1)$ representing the response of a binary scrambler to an input of period s, when s and $2^m - 1$ are relatively prime, has at least one transition every $s + m$ digits and has a total of $Tr(\boldsymbol{l})$ transitions where

$$\frac{1}{2}\left(\frac{2^m - 2}{2^m - 1}\right) \leqq \frac{\mathrm{Tr}\,(\boldsymbol{l})}{s(2^m - 1)} \leqq \frac{1}{2}\left(\frac{2^m}{2^m - 1}\right). \tag{39}$$

We begin by showing that every set of $s + m$ consecutive line digits must contain at least one transition. The scrambled sequence is the response of the basic scrambler of Fig. 2 to an input of period s. We note that if the basic scrambler is in the all zero state then the tap sum (which is added to the data bit) is zero. Similarly, if it is in the all 1 state the tap sum is zero because if not, $h(1) = 0$ and $h(x)$ is divisible by $x - 1$ which is impossible since $h(x)$ is irreducible. Then, if $s + m$ consecutive outputs of the scrambler are identical, the last s of the $(s + m)$ corresponding tap sums are zero so that s consecutive data bits must be identical. This cannot happen if the source is periodic with period greater than 1. When $s = 1$, the line sequence must have period 1 if $s + m$ consecutive line digits are identical, which also cannot happen since the line sequence has period $2^m - 1$ in this case.

We now bound Tr $(\boldsymbol{l})$, the number of transitions in one period, $\boldsymbol{l}$, of the line sequence. We use the notation of Section V so that the jth digit of $\boldsymbol{l}$, namely l_j is written

$$l_j = [T^j \mathbf{y}]_* , \tag{40}$$

where T is given by (3) through (5) and $\mathbf{y}$ is the state of the driven

basic scrambler at the beginning of a period of the data sequence. Let us now observe that a *transition occurs in between two digits in $\mathbf{l}$ if they sum to* 1 *modulo* 2. Thus, the number of 1's in $\mathbf{l} + \mathbf{l}'$ (where $\mathbf{l}'$ is one cyclic shift of $\mathbf{l}$ and addition is term-by-term) is the number of transitions in $\mathbf{l}$. For example, if $\mathbf{l} = 10110$, $\mathbf{l}' = 01011$ and $\mathbf{l} + \mathbf{l}' = 11101$ then the number of transitions in $\mathbf{l}$, including the implicit transition at the first digit is the Hamming weight of $\mathbf{l} + \mathbf{l}'$.

In the process of proving Theorem 4 we have shown (see (35)) that the Hamming weight of one period of the output of the basic scrambler when the input is binary of period s_0 and s_0 and $2^m - 1$ are relatively prime lies between $s_0(2^{m-1} - 1)$ and $s_0 2^{m-1}$. Hence, if we can show that $\mathbf{l} + \mathbf{l}'$ is one period of the output of the scrambler with input period s_0, we will have established Theorem 5.

We note that

$$l_i + l_i' = [T^i \mathbf{y} + T^{i-1}\mathbf{y}], \tag{41}$$

so that we now examine $T^{i+1}\mathbf{y} + T^i\mathbf{y}$. We have

$$T^i\mathbf{y} + T^{i-1}\mathbf{y} = T^{i-1}(T + I)\mathbf{y}. \tag{42}$$

As in (7), let $\mathbf{y} = \mathbf{y}_s + \mathbf{y}_m$ where $\mathbf{y}_s$ is zero in its last m components, $\mathbf{y}_m$ is zero in its first s components and they represent the periodic input and starting state of the basic scrambler, respectively. Then,

$$(T + I)\mathbf{y} = \mathbf{y}_s + \mathbf{y}_s' + (\underbrace{0}_{s}, y_s, \underbrace{0}_{m-1}) + \mathbf{y}_m + T\mathbf{y}_m, \tag{43}$$

where $\mathbf{y}_s'$ is a single cyclic shift of $\mathbf{y}_s$ in its first s places and $(\underbrace{0}_{s}, y_s, \underbrace{0}_{m-1})$ is a vector with a single component y_s in the $(s + 1)$st position. If we use $(\mathbf{z})'$ to represent the last m components of $\mathbf{z}$, then

$$(\mathbf{y}_m + T\mathbf{y}_m)' = (T_h + I_m)(\mathbf{y}_m)'. \tag{44}$$

In the appendix it has been established that $T_h + I_m$ is a nonsingular matrix. From this we deduce that the last m components of $(T + I)\mathbf{y}$ range over all 2^m m-tuples as $\mathbf{y}_m$ ranges over all m-tuples.

Now consider $\mathbf{y}_s + \mathbf{y}_s'$, which represents the first s components of $(T + I)\mathbf{y}$. While $\mathbf{y}_s$ models one period of a data sequence with period exactly s, $\mathbf{y}_s + \mathbf{y}_s'$ may model a sequence with period s_0, $s_0 \mid s$. For example, let $\mathbf{y}_4 = (1001000)$, then $\mathbf{y}_4 + \mathbf{y}_4' = (0101000)$ and its first 4 components represent two periods of a period 2 sequence. Thus, we must view $(T + I)\mathbf{y}$ as the starting state of a driven basic scrambler with input period s_0 where $s_0 \mid s$. We then ask if the sequence generated by this state has period s_0 or $s_0(2^m - 1)$. Since $\mathbf{y}$ is noncritical, the

sequence generated by $(T^t + I)\mathbf{y}$ must have tbe larger period because if $\mathbf{y}$ were critical $(T^i + I)\mathbf{y} = 0$ for some i, $1 \leqq i \leqq s_0(2^m - 1) - 1$ and $(T^i + I)(T + I)\mathbf{y} = 0$ as well for some i in this range so tbat $(T + I)\mathbf{y}$ is a critical state. But we have sbown in Theorem 1 tbat tbere is only one critical state for eacb periodic input. In the last paragraph we have seen that there is a one-to-one mapping between tbe last m components of $\mathbf{y}$ and the last m components of $(T + I)\mathbf{y}$, hence if $\mathbf{y}$ is noncritical, $(T + I)\mathbf{y}$ is noncritical and the line sequence generated by $(T + I)\mathbf{y}$ has period $s_0(2^m - 1)$ where $s_0 \mid s$.

The vector $l + l'$ contains s/s_0 periods of a sequence of period s_0. Let $\mathbf{C}_n$ represent one such period. Then from (35), the number of 1's in $\mathbf{C}_n$, $w(\mathbf{C}_n)$, is bounded by

$$s_0(2^{m-1} - 1) \leqq w(\mathbf{C}_n) \leqq s_0 2^{m-1}. \tag{35}$$

Then,

$$\mathrm{Tr}\,(l) = w(l + l')$$

and

$$s(2^{m-1} - 1) \leqq \mathrm{Tr}\,(l) \leqq s2^{m-1} \tag{45}$$

which gives the desired result after division by $s(2^m - 1)$.

X. THE SELF-SYNCHRONIZING DESCRAMBLERS

In this section, we show that the descrambler for each of the scramblers given above has the self-synchronizing property, that it is relatively insensitive to channel errors and tbat in some applications it can be considerably simplified by removal of the monitoring logic.

Eacb scrambler is of the form shown in Fig. 9. Each descrambler can be represented as shown in Fig. 10. Tbe output marked "data" in Fig. 10 is indeed data if the scrambler and descrambler are both started in the same state and no channel errors occur since (*i*) the line sequence will then pass through both basic scramblers and (*ii*) the modulo p sum of a data bit, tap sum, line bit, and monitoring logic output is zero at both the scrambler and descrambler.

If there are no channel errors we would like to show that the descrambler will synchronize itself should it ever lose synchronism. The descrambler will be said to be out of synchronism with the scrambler if either the values stored in the basic scrambler and the delay elements differ from those stored in corresponding sections of the scrambler or if the counters in the monitoring logic are not at the same levels as those at the scrambler or both. It is clear that the s_N stages (if the largest

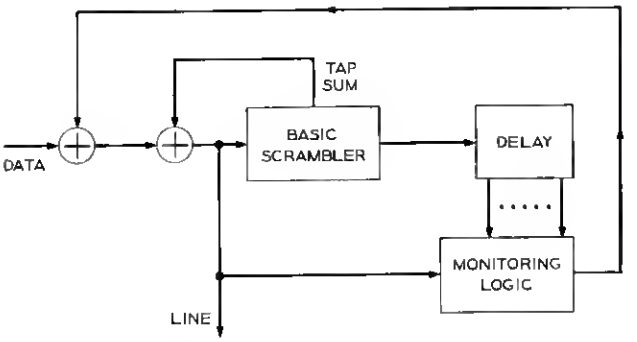

Fig. 9—Block diagram of the scrambler.

expected period is s_N) of the basic scrambler in the descrambler and delay section will be purged after s_N clock intervals and replaced with accurate information if there are no channel errors. Then, after s_N clock intervals the monitoring logic at the scrambler and descrambler both are fed the same information. The monitoring logics will then reach synchronism when either (*i*) counters at the scrambler and descrambler reach threshold together in which case all counters are reset simultaneously or (*ii*) the last $s_N + 1$ digits of the line sequence is found to be inconsistent with a periodic sequence of period s_1, s_2, $\cdots$ or s_N and the counters at the descrambler are reset individually but in synchronism with those at the scrambler. When the data sequence is *periodic* of period s_1, s_2, $\cdots$ or s_N the ith counter of the MCS is reset (following the transient interval associated with the arrival of

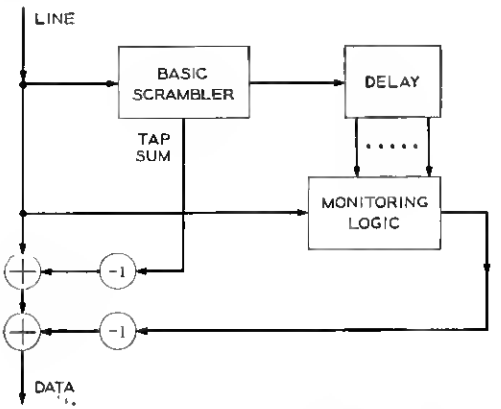

Fig. 10—Block diagram of the descrambler.

the periodic sequence) at least once every $t_{s_i} = m - 1 + \max_{j \neq i} s_j$ clock intervals. With the SCS the single counter is reset at least once every $s_2(2^m - 1) - 2^{m-1} + 2$ clock intervals when the input has period s_1 or s_2, $s_1 < s_2$. Should the input sequence be *random*, the monitoring logics may be brought into synchronism because one of the counters reaches threshold and all counters are reset, which is unlikely, or because the counters are reset individually in synchronism with the scrambler counters, which is very probable and increases in probability very rapidly to one. (If the source is binary with independent, equiprobable outputs, the ith counter of the MCS descrambler is resynchronized in the second manner after n clock intervals with probability $1 - 2^{-n}$; similarly, the counter of the SCS descrambler is resynchronized with probability $1 - (\frac{3}{4})^n$.)

Channel errors can affect the process of resynchronization. However, if we assume that they are relatively few in number, say, occurring once in every 10^5 transmissions, there will be long intervals during which resynchronization can take place. Since the descrambler requires at most $s_N + \max t_{s_i}$ (which equals $2s_N + m - 1$ in the MCS case when $s_i \leqq s_N$ and is at most $s_2 2^m - 2^{m-1} + 2$ in the SCS case) clock intervals to resynchronize when the source is periodic, resynchronization will not be a problem with periodic inputs if m and s_N (or s_2) are reasonable in size. When the source output is random and is a sequence of independent, equiprobable binary digits, the average number of clock cycles required by the ith counter of the MCS descrambler to resynchronize (in the second way described in the preceding paragraph) is two so that the MCS descrambler will resynchronize on the average in $s_N + 2$ clock intervals. The counter of the SCS descrambler will require four clock intervals on the average to resynchronize so that the SCS descrambler will be resynchronized on the average in $s_N + 4$ clock intervals. Hence, we may conclude that resynchronization in the presence of channel errors which are relatively few in number will not be a problem when the source is random. In fact, it may be easier to resynchronize when the data is random than it is when the data is periodic.

Now assume that the scrambler and descrambler are operating in synchronism and consider the effect of channel errors on the descrambler output. If we neglect the monitoring logic for a moment, it will be be seen that an isolated channel error, as it passes through the basic scrambler, will cause $w(h)$ output errors, where $w(h)$ is the number of nonzero terms in the tap polynomial $h(x)$. The monitoring logic, however, may fail to act when it should or act when it should not and thereby

introduce additional errors. If we consider the effect of a single channel error on the monitoring logic, we see that this error has a direct effect on the ith counter of the MCS at two occasions, when it enters the basic scrambler and when it reaches the s_ith storage element. A single channel error has a direct effect on the counter of the SCS three times, once when it enters the basic scrambler and again when it enters the s_1th and s_2th storage elements. When the channel error effects a counter of the descrambler, it may cause it to reset when it should not, which will not cause any harm if the counter is about to be reset before reaching threshold, as is the case for the known periodic inputs or as frequently happens with a random source. A channel error which causes a counter to continue to count when it should reset may indeed be harmful since it may result in its reaching threshold and introduce an unnecessary change in the descrambler output. This event is unlikely to happen for the known periodic source sequences since the counters reset frequently, and the number of clock intervals between a set of three normal counter resets is often less than a given counter threshold. It is also unlikely that a channel error will eliminate a reset and cause a counter to reach threshold when the source is random. For example, when the source is a binary, equiprobable, independent letter source the average separation between three resets on the MCS counters is four clock intervals and is eight clock intervals on the SCS. We may conclude then *that channel errors have a small effect on the monitoring logic and thus affect the descrambler primarily by producing approximately $w(h)$ as many output errors as channel errors.*

The descrambler can be considerably simplified, the problem of synchronization loss in the descrambler monitoring logic eliminated, and the problem of output errors due to the monitoring logic solved, all by the removal of the monitoring logic at the descrambler. This is not the drastic solution that it might seem for the monitoring logic reacts infrequently on random data and at most twice on known periodic inputs (if counter thresholds all are larger than the largest expected input period). With a binary, independent, equiprobable letter source, one or more of the N counters of the MCS reaches threshold in n transmissions with a probability, $P_M(n)$, which is less than or equal to

$$P_M(n) \leqq \sum_{i=1}^{N} (n - t_i + 1)2^{-t_i} , \tag{46}$$

where t_i is the threshold on the ith counter and $t_i \geqq t_{i-1}$. The single counter of the SCS reaches threshold t in n transmissions with prob-

ability $P_S(n)$ where

$$P_S(n) \leqq (n - t + 1)\left(\tfrac{3}{4}\right)^{-t}. \tag{47}$$

Hence, if the thresholds are large enough so that $P_M(n)$ or $P_*(n)$ is less than 0.1, say, when n equals the average number of transmissions between channel errors, then we may safely say that the monitoring logic at the descrambler is not necessary on random data inputs.

When the source is periodic of period s however, one of the p^m starting states* of the basic scrambler will result in a line sequence of period s which subsequently will require at least one and at most two outputs from the monitoring logic. Thus, if the data preceding a periodic input is random, the monitoring logic at the descrambler will with probability $1/p^m$ change at least 1 digit in the descrambler output. Hence, if a customer can tolerate such an error rate and if the thresholds are large enough, the monitoring logic at the descrambler can be removed and the descrambler will then simply consist of a basic scrambler.

XI. THE SPECTRUM OF THE SCRAMBLER OUTPUT

In this section, we perform representative calculations to show the effect of scrambling on the spectrum of a linearly modulated carrier. Assume that the source is binary and that a binary sequence is converted into a waveform by the mapping $0 \to -1$, $1 \to +1$. Let T_0 be the time interval alloted to each binary digit and let $\hat{l}(t)$ be the waveform generated by the binary sequence $\boldsymbol{l}$. Then, we have

$$\hat{l}_1(t) \cdot \hat{l}_2(t) = -\widehat{(\boldsymbol{l}_1 + \boldsymbol{l}_2)}(t) \tag{48}$$

where addition is taken modulo 2 and multiplication is on the reals.

The autocorrelation function of a waveform $\hat{l}(t)$ is defined as

$$R_l(\tau) = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} \hat{l}(t) l(t + \tau)\, dt. \tag{49}$$

If $\boldsymbol{l}$ is the output of the scrambler when the input is an equiprobable, independent letter source, then $\boldsymbol{l}$ is a sequence of independent, equiprobable, binary digits. Then, we have

$$R_l(\tau) = \begin{cases} \left(1 - \dfrac{|\tau|}{T_0}\right) & |\tau| \leqq T_0, \\ 0 & |\tau| > T_0. \end{cases} \tag{50}$$

* p is the input alphabet size and m is the number of stages in the basic scrambler.

The power density spectrum of $\hat{l}(t)$, which is the Fourier Transform of $R_l(\tau)$ is for the random binary source

$$S_l(f) = T_0\left(\frac{\sin \pi f T_0}{\pi f T_0}\right)^2. \tag{51}$$

Now let the source be periodic and assume, as an example, that it has period 8 and that the following sequence is one period of the source output: **10110010**. Then, if l represents this sequence and if it is transmitted without scrambling, we find using (48) that it has the autocorrelation function, $R_l(\tau)$, of Fig. 11. The power density spectrum of $\hat{l}(t)$, $S_l(f)$, is given below and shown in Fig. 12.

$$S_l(f) = 2T_0\left[\left(\frac{\sin \pi f T_0}{\pi f T_0}\right)^2 - \left(\frac{\sin 2\pi f T_0}{2\pi f T_0}\right)^2\right] \sum_{j=-\infty}^{\infty} \frac{1}{8T_0}\,\delta\left(f - \frac{j}{8T_0}\right). \tag{52}$$

Here $\delta(\cdot)$ is the Dirac delta function. Thus, $S_l(f)$ contains isolated tones spaced by $1/T_1$, $T_1 = 8T_0$, the period of the data sequence.

If the periodic data source of period s is now scrambled, the line sequence has period $T_0(LCM\ (s, 2^m - 1))$. Assume now, as an example, that s and $2^m - 1$ are relatively prime so that the line sequence has period PT_1, $P = 2^m - 1$, the scale-up factor, and $T_1 = sT_0$, the source period. Now let $\boldsymbol{l}$ represent *one period* of the binary line sequence. Then, if $\boldsymbol{l}_k$ represents k cyclic shifts of $\boldsymbol{l}$ we have

$$R_l(kT_0) = \frac{1}{PT_1}\int_{-PT_1/2}^{PT_1/2} \hat{l}(t)\hat{l}_k(t)\,dt. \tag{53}$$

When $k = \pm 1, \pm 2, \cdots, \pm(P-1)$, we have

$$R_l(kT_0) = -\frac{T_0}{PT_1}\,(\text{No. 1's in } (\boldsymbol{l} + \boldsymbol{l}_k) - \text{No. 0's in } \boldsymbol{l} + \boldsymbol{l}_k). \tag{54}$$

Since $R_l(\tau)$ is linear in τ for $(k-1)T_0 \leqq \tau \leqq kT_0$ we need only have $R_l(\tau)$ at $\tau = kT_0$, $k = 0, \pm 1, \pm 2, \cdots$. We note that $R_l(kPT_1) = 1$,

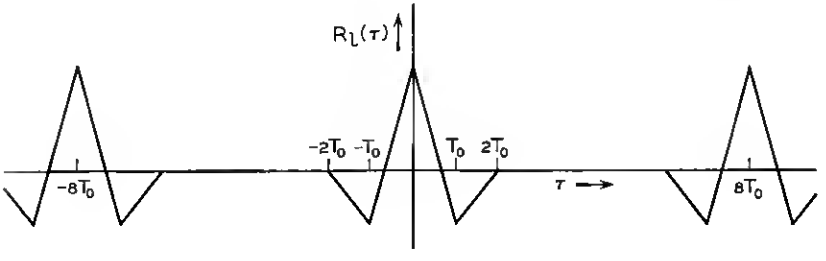

Fig. 11 — Autocorrelation function of period 8 sequence.

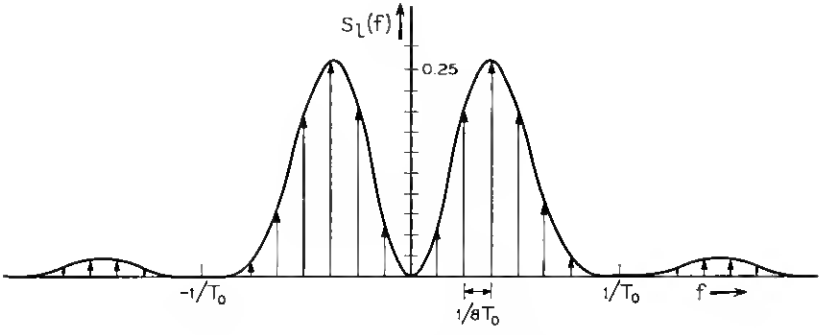

Fig. 12—Spectrum of period 8 sequence.

$k = 0, \pm 1, \pm 2, \cdots$. To further evaluate (54), however, we must return to Section IX.

We have seen in Section IX that $l + l_1$ represents several periods in the output of a basic scrambler driven by an input of period s_0 , $s_0 \mid s$, and started with a noncritical state $\mathbf{y}$. The proof of this result amounted to showing that the operation $(T + I)$ on $\mathbf{y}$ mapped the last m components of $\mathbf{y}$ one-to-one onto the last m components of $(T + I)\mathbf{y}$. Thus, if $\mathbf{y}$ is critical so is $(T + I)\mathbf{y}$ and since there is only one critical state for each periodic input $(T + I)\mathbf{y}$ is noncritical if $\mathbf{y}$ is noncritical. We can show in a similar manner that $(T^k + I)\mathbf{y}$ is noncritical when $\mathbf{y}$ is noncritical as long as k is not a multiple of $2^m - 1$. Thus, $l + l_k$, which is produced by the starting state $(T^k + I)\mathbf{y}$ when $\mathbf{y}$ generates l, is the output of a basic scrambler with input period s_0 and output period $s_0(2^m - 1)$ when k is not a multiple of $2^m - 1$ and $s_0 \mid s$. Then, invoking (35), we have

$$-1/P \leqq R_l(kT_0) \leqq 1/P, \; k \text{ not a multiple of } P = 2^m - 1. \tag{55}$$

We note, however, that $R_l(kT_0)$ for such k may not all be equal.

Next consider $l + l_k$ when k is a multiple of $2^m - 1$. If $l + l_k$ represents an output which has period which divides s, then $(T^s + I)(T^k + I)\mathbf{y} = 0$. We now show that $(T^s + I)(T^k + I) = 0$ for all $\mathbf{y}$ when k is a multiple of $2^m - 1$. We observe that

$$T^s + I = \begin{bmatrix} \underline{0} & \underline{0} \\ \hline Q_s & T_k^s + I_m \end{bmatrix} \tag{56}$$

and

$$T^k + I = \begin{bmatrix} R^k + I_s & \underline{0} \\ \hline Q_k & \underline{0} \end{bmatrix} \tag{57}$$

since $T_k^k = I$. Because $T^s + I$ and $T^k + I$ commute, we have

$$(T^s + I)(T^k + I) = (T^k + I)(T^s + I) = \underline{0}. \tag{58}$$

Thus, $\boldsymbol{l} + \boldsymbol{l}_k$ represents a scramhler output of period s_0 , where $s_0 \mid s$. It is clear then that the numher of 1's in one period of $\boldsymbol{l} + \boldsymbol{l}_k$ is greater than or equal to 1 and less than or equal to $s_0 - 1$. Also, $\boldsymbol{l} + \boldsymbol{l}_k$ is the same sequence for all multiples of $P = 2^m - 1$ which are not multiples of sP. Thus, for k a multiple of P which is not a multiple of sP, we have from (52) that

$$-\frac{(s-2)}{s} \leqq -\left(\frac{s_0 - 2}{s_0}\right) \leqq R_l(kT_0) \leqq \left(\frac{s_0 - 2}{s_0}\right) \leqq \frac{s-2}{s} \tag{59}$$

when $s_0 \geqq 2$.

To calculate a representative spectrum of the scrambled data sequence, we assume that $R_l(\tau)$ has the following form, where u, $2 \leqq u \leqq 2s$, is a function of the scramhler input (the numher of 1's in $\boldsymbol{l} + \boldsymbol{l}_k$, k a multiple of P, depends on the input):

$$R_l(kT_0) = \begin{cases} 1 & k = nsP, \quad n = 0, \quad \pm 1, \pm 2, \cdots, \\ \dfrac{s-u}{s} & k = nP, \quad n \neq 0, \quad \pm s, \pm 2s, \cdots, \\ \dfrac{1}{P} & \text{all other } k. \end{cases} \tag{60}$$

The power density spectrum $S_l(f)$ then is

$$S_l(f) = \frac{1}{P}\,\delta(f) + T_0\left(\frac{\sin \pi f T_0}{\pi f T_0}\right)^2 \left\{ \frac{u}{SPT_1} \sum_{j=-\infty}^{\infty} \delta\left(f - \frac{j}{PT_1}\right) + \left(1 - \frac{u}{s} - \frac{1}{P}\right) \frac{1}{PT_0} \sum_{j=-\infty}^{\infty} \delta\left(f - \frac{j}{PT_0}\right) \right\}. \tag{60}$$

When u is of the order of s we see that the second term in curly brackets has amplitudes which are proportional to $1/P^2$ and are thus much smaller than terms in the first sum. We show $R_l(\tau)$ with $u = s$, $R_l(T_0) = \epsilon$ in Fig. 13 and $S_l(f)$ in Fig. 14. The assumption that $u = s$ is equivalent to the assumption that $\boldsymbol{l} + \boldsymbol{l}_k$ contains an equal numher of 1's and 0's when k is a multiple of P.

We deduce from this discussion of spectra that *the principal effect of scrambling* when the scrambled sequence is converted to a signal waveform in the manner given ahove *is to increase the number of tones in a given bandwidth by a factor which is approximately P and to decrease the level of each tone by approximately the same factor.*

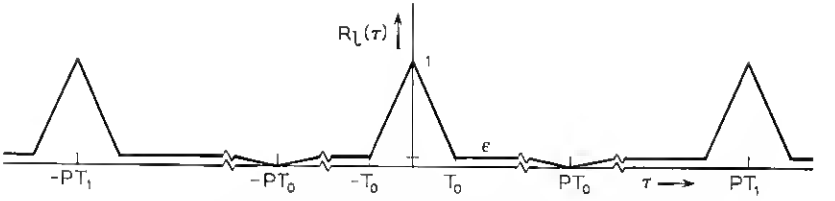

Fig. 13 — Autocorrelation function of a scrambled periodic sequence.

XII. AN EXAMPLE

We shall now consider an application for the scramblers and we shall compare the relative cost and effectiveness of the MCS and the SCS. We will report on a computer simulation directed at the determination of the smallest SCS counter threshold for our example.

Assume that the source is binary and that it may occasionally contain sequences of period 1 (there are two—the all 0 sequence and the all 1 sequence), period 2 (there is only one—the 1010 $\cdots$ sequence, known as dotting), period 7 or period 8. Assume also that a line sequence of period less than 100 is undesirable from spectra considerations. Since the least common multiple of 1 and $2^m - 1$ is $2^m - 1$, we will require that $2^m - 1 > 100$. The smallest value of m for which this is true is $m = 7$ for which $2^m - 1 = 127$, a prime. We next require a primitive, degree 7 binary polynomial for the tap polynomial. The polynomial $h(x) = 1 + x^4 + x^7$ is one such. Given $h(x)$ our basic scrambler is fixed. We next observe that 1 divides 7 and 8 and that 2 divides 8 so that we may build a scrambler which detects two periods $s_1 = 7$ and $s_2 = 8$.

We next consider whether the MCS or the SCS should be used for our problem. We see immediately from Theorem 3 that the threshold

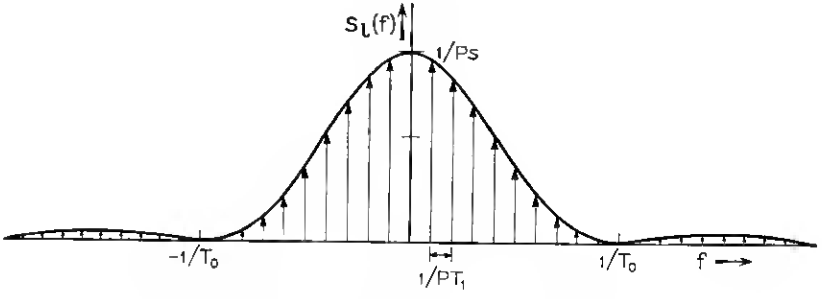

Fig. 14 — Spectrum of a scrambled periodic sequence.

on the first counter of the MCS, t_{s_1}, must be at least 14. Similarly, t_{s_2} of the MCS must be at least 13. Since a 4-stage binary counter will count to 16, we see directly that 8 counter stages, 8 shift register stages, 3 OR gates and some peripheral logic will suffice to build an MCS for our problem.

From Theorem 4 we see that the threshold on the SCS need not be any larger than 954 or require more than 10 stages of a binary counter since $2^{10} = 1024 > 954$. A computer simulation of the SCS, however, shows that the bound of 954 is more than 34 times larger than the smallest required threshold, which was found to be 28. The results of this simulation are tabulated in Table I. The largest run of consecutive zeros at counter input was found for all period 7 and period 8 sequences when the line sequences had periods $7 \cdot 127$ and $8 \cdot 127$, respectively. In Table I we list the fraction of the 384 periodic sequences which have the gap lengths (maximum run of zeros) shown. (We note that it is only necessary to simulate the SCS with one starting state of the basic scrambler when $2^m - 1$ is prime since all $2^m - 1$ noncritical starting states appear as states of the basic scrambler. Note also that we can neglect the first eight inputs to the counter following the argument of the third paragraph of Section VII.)

The SCS will scramble our periodic inputs if we choose a counter threshold of 32 which can be realized with a 5-stage binary counter. It will also require eight shift register stages, an AND gate and peripheral logic.

As far as random data is concerned, we see from (46) that the MCS

TABLE I—GAP LENGTHS FOR PERIODIC INPUTS

	Period 7		Period 8	
Gap length	No.	%	No.	%
13	14	10.92	0	0
14	28	21.84	16	6.25
15	14	10.92	16	6.25
16	0	0	16	6.25
17	0	0	16	6.25
18	28	21.84	60	23.41
19	14	10.92	18	7.04
21	0	0	56	21.85
22	2	1.56	2	0.78
24	0	0	24	9.38
25	0	0	16	6.25
26	14	10.92	0	0
27	14	10.92	16	6.25
	128		256	

for our application reaches threshold at least once in n transmissions with probability

$$P_M(n) \leqq (n - 15)(3.06)10^{-5} \tag{61}$$

and we see from (47) that the SCS reaches threshold with probability

$$P_S(n) \leqq (n - 31)10^{-4}. \tag{62}$$

Thus, the MCS has a slight edge on the SCS when it comes to scrambling random data since it is desirable to keep the frequency of threshold crossings low.

In sum, it is safe to say that the SCS has the edge for our problem primarily because it is simpler and less expensive. Also, we note that the addition of a single-counter stage will reduce $P_S(n)$ to $(n - 31)10^{-8}$. The autocorrelation function of the scrambled data sequence will be like that of Fig. 13 with $| \epsilon | \leqq 0.008$.

XIII. CONCLUSIONS

We have introduced two major classes of self-synchronizing, digital data scramblers called multi-counter scramblers and single-counter scramblers. We have shown that these scramblers and combinations of the two will map a periodic sequence of period s into a periodic sequence of period $LCM(s, p^m - 1)$, where p is the size of the source alphabet (the SCS results require that $p = 2$ and that s and $2^m - 1$ be relatively prime), if the basic scrambler tap polynomial $h(x)$ of degree m is a primitive polynomial over $GF(p)$. We have found the smallest values for the counter thresholds in the MCS and have shown the existence of finite thresholds for the successful operation of the SCS.

We have shown that there are many transitions in the scrambled sequence and that they are well distributed. We have shown that the descramblers possess the self-synchroniziug property and we have considered the effect of channel errors on the descrambling process. We have seen that the principal effect of infrequent channel errors (occurring at a rate of one in 10^5 transmissions, say) is to cause approximately $w(h)$ as many output errors, where $w(h)$ is the number of nonzero terms in $h(x)$. Channel errors were shown to have a relatively small effect on the output of the descrambler monitoring logic.

We have found the power density spectrum of the waveform generated by the scrambler output for a representative case, namely, when the source is binary and the scrambled sequence is mapped onto a ± 1 sequence. We have seen that scrambling does not affect the spectrum

of the line signal when the source is random and that its principal effect when the source is periodic is to introduce P times as many tones each having $1/P$th as much energy where P is the factor by which the source period is increased.

It has been shown that the counters in the scrambler and descrambler reach threshold infrequently when the source is random and at most once each time the source becomes periodic. Thus, it has been argued that the counters at the descrambler might be removed if the rate at which the counters at the scrambler reach threshold is less than the rate of occurrence of channel errors, and if the customer can tolerate occasional output errors when his data is periodic.

XIV. ACKNOWLEDGMENTS

The author acknowledges helpful early conversations with R. D. Fracassi in this work and many discussions with his colleagues in the Data Communications Laboratory.

APPENDIX

Proof of Theorem 2

Let T_h be the matrix shown below where the coefficients c_1, c_2, $\cdots$, c_m are elements of the modular field $GF(p)$ of p elements, p a prime

$$T_h = \begin{bmatrix} c_1 & c_2 & \cdots & c_{m-1} & c_m \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \uparrow \downarrow m. \tag{63}$$

Let $h(x)$ be the polynomial shown below in the indeterminate x where coefficients are those appearing in (63).

$$h(x) = x^m - c_1 x^{m-1} - \cdots - c_m . \tag{64}$$

Then, one can show by direct calculation that the characteristic polynomial of T_h, $\varphi(x)$, defined by

$$\varphi(x) = \det (T_h - xI), \tag{65}$$

is related to $h(x)$[9,10] by

$$\varphi(x) = (-1)^m h(x). \tag{66}$$

The matrix T_h is called the "companion matrix" for the polynomial $h(x)$.

We assume that $h(x)$ is a primitive polynomial over the field $GF(p)$. A polynomial $h(x)$ is primitive if

(*i*) $h(x)$ is irreducible over $GF(p)$, that is, if there is no polynomial with coefficients in $GF(p)$ which divides $h(x)$ except 1 and $h(x)$, itself, and

(*ii*) $h(x)$ of degree m divides $x^n - 1$ for $n = p^m - 1$ but for no smaller integer n.

If we replace the term c_m in $h(x)$ given in (64) by the matrix $c_m I$, where I is the $m \times m$ identity matrix and replace x by T_h, where powers of T_h are defined as successive matrix products, then we have the well-known Cayley-Hamilton theorem[11]

$$\varphi(T_h) = 0, \tag{67}$$

where $\varphi(x)$ is the characteristic polynomial of T_h. Thus, a matrix T_h satisfies its own characteristic polynomial. There is a smallest degree monic polynomial (coefficient of the highest degree term is 1), called the minimal polynomial, $m(x)$, such that

$$m(T_h) = 0. \tag{68}$$

Since $h(x)$ is irreducible, we have

$$m(x) = h(x). \tag{69}$$

We now wish to prove the following theorem.

Theorem 2: The matrices $T_h^k - I$ are nonsingular for $1 \leqq k \leqq p^m - 2$.

We first prove the following two lemmas.

Lemma 2: If $0 \leqq i, j \leqq p^m - 2$, $i \neq j$, then $T_h^i \neq T_h^j$.

Proof: If $T^i = T^j$ for the i, j given above and $i < j$ then

$$T^i(T^{j-i} - I) = 0$$

implies

$$T^{j-i} - I = 0$$

since $\det T_h = \varphi(0) \neq 0$. (If $\varphi(0) = 0$ then $\varphi(x)$ is divisible by x and $h(x)$ is not primitive.) Consider now the polynomial $x^{j-i} - 1$. Using the Euclidean division algorithm we have

$$x^{j-1} - 1 = h(x)q(x) + s(x)$$

for unique $q(x)$ and $s(x)$ and degree $s(x) <$ degree $h(x)$. Therefore,

$$T_h^{i-1} - I = 0 = h(T_h)q(T_h) + s(T_h)$$

which implies that

$$s(T_h) = 0.$$

But $m(x) = h(x)$ is the minimal polynomial of T_h so that $s(x) = 0$. Therefore, $h(x)$ divides $x^n - 1$, $n = j - i < p^m - 1$. Contradiction. Hence, $T^i \neq T^j$, $i \neq j$, $0 \leqq i, j \leqq p^m - 2$. QED

Lemma 3: All nonzero polynomials in T_h with coefficients in $GF(p)$ and of degree $m - 1$ or less are nonsingular.

Proof: Let $p(x)$ be a polynomial of degree $m - 1$ or less with coefficients in $GF(p)$. Then, using the Euclidean division algorithm, we have that the greatest common divisor, $d(x)$, of $p(x)$ and $h(x)$ is given by

$$d(x) = a(x)p(x) + b(x)h(x),$$

where $a(x)$ and $b(x)$ are unique polynomials. Since $h(x)$ has degree m and is irreducible $d(x) = 1$ and

$$1 = a(x)p(x) + b(x)h(x).$$

Taking these polynomials in T_h, we have

$$I = a(T_h)p(T_h) + b(T_h)h(T_h)$$

or since $h(T_h) = 0$ we have

$$I = a(T_h)p(T_h) = p(T_h)a(T_h),$$

where the latter equality follows since the polynomials $a(x)$ and $p(x)$ commute. Thus, the polynomial $p(T_h)$ of degree $m - 1$ or less with coefficients over $GF(p)$ in the matrix T_h has both a left inverse and a right inverse and is nonsingular. QED

Proof of Theorem 2:

Since $h(T_h) = 0$ we have

$$T_h^m = c_1 T_h^{m-1} + c_2 T_h^{m-2} + \cdots + c_m I.$$

Thus, every power of T_h, such as T_h^i can be written as a polynomial in T_h of degree $m - 1$ or less. Hence, $T_h^i - T_h^j$ can be written as a polynomial of degree $m - 1$ or less in T_h. From Lemma A1, $T_h^i - T_h^j \neq 0$, $i \neq j$, $0 \leqq i, j \leqq p^m - 2$ so that $T_h^i - T_h^j$ as a polynomial in T_h of degree $m - 1$ or less is nonzero. From Lemma A2, $T_h^i - T_h^j$ is nonsingular

and it follows by choosing $j = 0$, $i = k$ with $1 \leqq k \leqq p^m - 2$ that $T_h^k - I$ is nonsingular. QED

Theorem 2 in effect says that if y is some arbitrary, nonzero column vector of m components chosen from $GF(p)$ then $T^k\mathbf{y}$ runs through all $p^m - 1$ nonzero vectors y as k ranges between 0 and $p^m - 2$. Thus, *the linear sequential filter* with feedback paths *described by* T_h *is* a *maximal-length sequence generator.* Elspas[5] comments that these results were noted by Zierler[2] and Golomb.[12]

REFERENCES

1. Golomb et al., *Digital Communications with Space Applications*, Prentice-Hall, New Jersey, 1964.
2. Zierler, N., Several Binary Sequence Generators, Lincoln Lab., MIT, Lexington, Mass., Tech. Rep. No. 95; September, 1956, also reprinted in (13).
3. Fracassi, et al., Patent Application, Case 8-1, Serial 482498, Filed August 25, 1965.
4. Huffman, D. A., The Synthesis of Linear Sequential Coding Networks, Proc. Third London Symp. on Information, Theory, September, 1955, pp. 77–95, also reprinted in *Linear Sequential Switching Circuits* (see 13).
5. Elspas, B., The Theory of Autonomous Linear Sequential Networks, IRE Trans. Circuit Theory, *CT-6*, No. 1, March, 1959, pp. 45–60, also reprinted in *Linear Sequential Switching Circuits* (see 13).
6. Zierler, N., Linear Recurring Sequences, SIAM Journal, *7*, March, 1959, pp. 31–48, also reprinted in *Linear Sequential Switching Circuits* (see 13).
7. Peterson, W. W., *Error-Correcting Codes*, MIT Press and John Wiley & Sons, 1961, p. 7.
8. Bose, R. C. and Kuebler, Jr., R. R., A Geometry of Binary Sequences Associated with Group Alphabets in Information Theory, Annals Math. Stat., *31*, March, 1960, pp. 113–139.
9. Albert, A. A., *Fundamental Concepts of Higher Algebra*, The University of Chicago Press, 1956, p. 86.
10. Birkhoff, G. and MacLane, S., *A Survey of Modern Algebra*, Macmillan, 1941, pp. 316–318.
11. *Ibid.*, pp. 319–321.
12. Golomb, S. W., Sequences with Randomness Properties, Glenn, L. Martin Company, Baltimore, Md., Final Rep. on Contract No. SC54–33611, dated June 14, 1955 and cited as Ref. 7 in Ref. 5 above.
13. Kautz, W. H. (ed.), *Linear Sequential Switching Circuits*, Holden-Day, San Francisco, 1965.